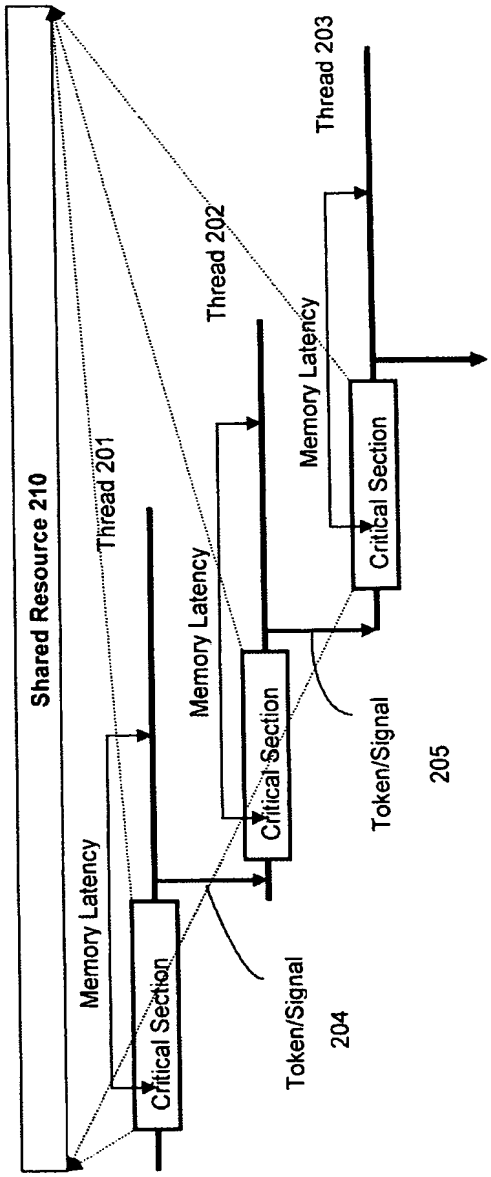

Shared Resource 210
Thread 201
Thread 202
Thread 203
Memory Latency
Memory Latency
Memory Latency
Critical Section
Critical Section
Critical Section
Token/Signal
Token/Signal
204
205

Fig. 2

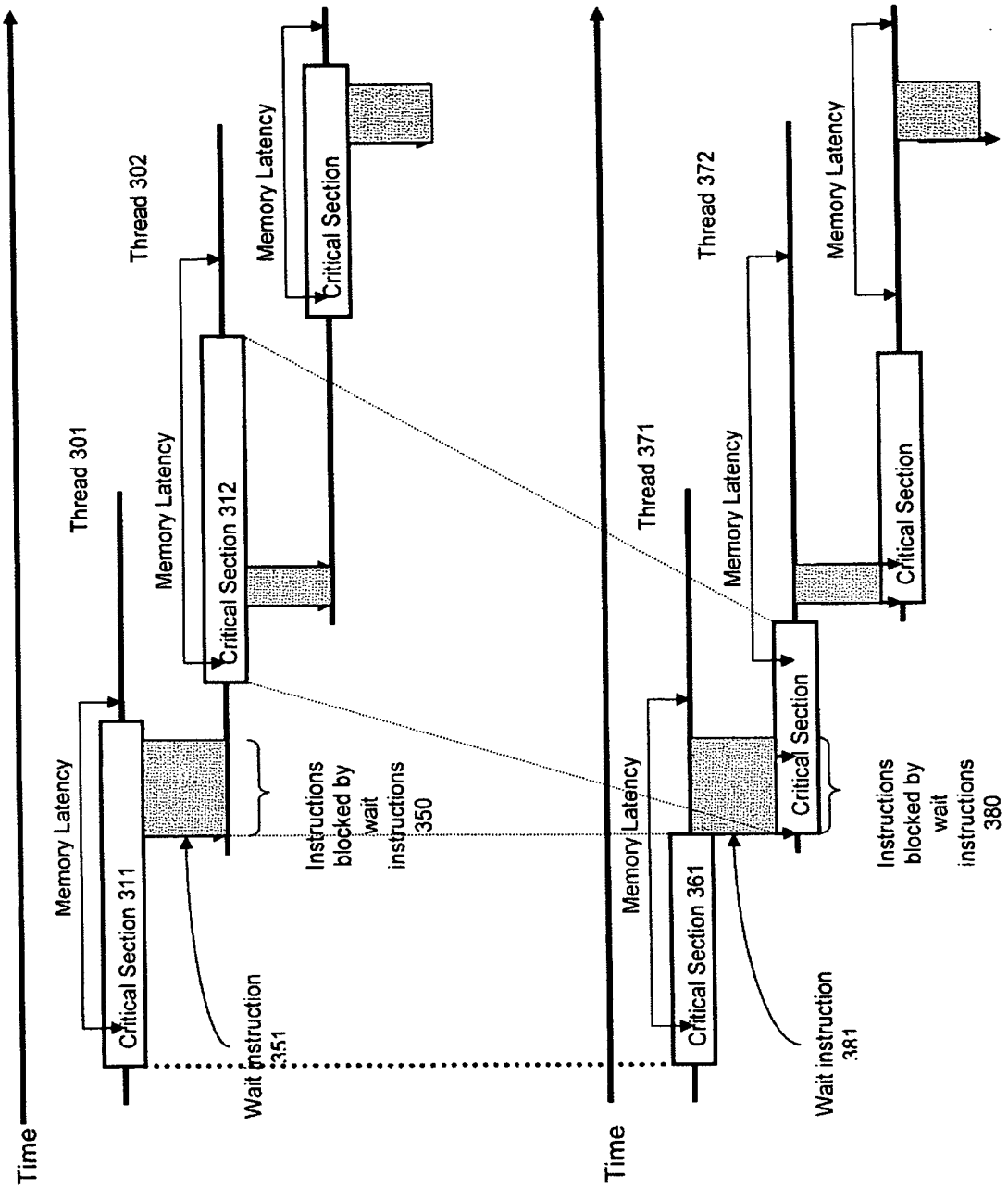
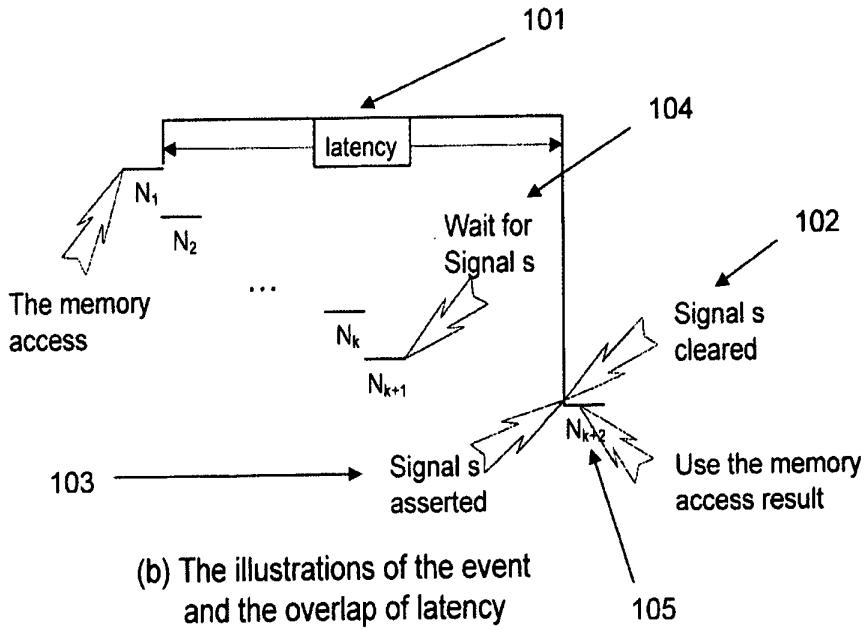

Time
Thread 301
Thread 302
Memory Latency
Critical Section 311
Critical Section 312
Critical Section
Wait instruction 351
Instructions blocked by wait instructions 350
Time
Thread 371
Thread 372
Memory Latency
Critical Section 361
Critical Section
Wait instruction 381
Instructions blocked by wait instructions 380

Fig. 3a

Fig. 3b

N_1: R1 = load [R2], signal s
N_2: R3 = R2 + 1
... ...
N_k: R4 = R3 << 2
N_{k+1}: wait s
N_{k+2}: R5 = R1 - R4

(a)



(b) The illustrations of the event and the overlap of latency

Figs. 1a-b

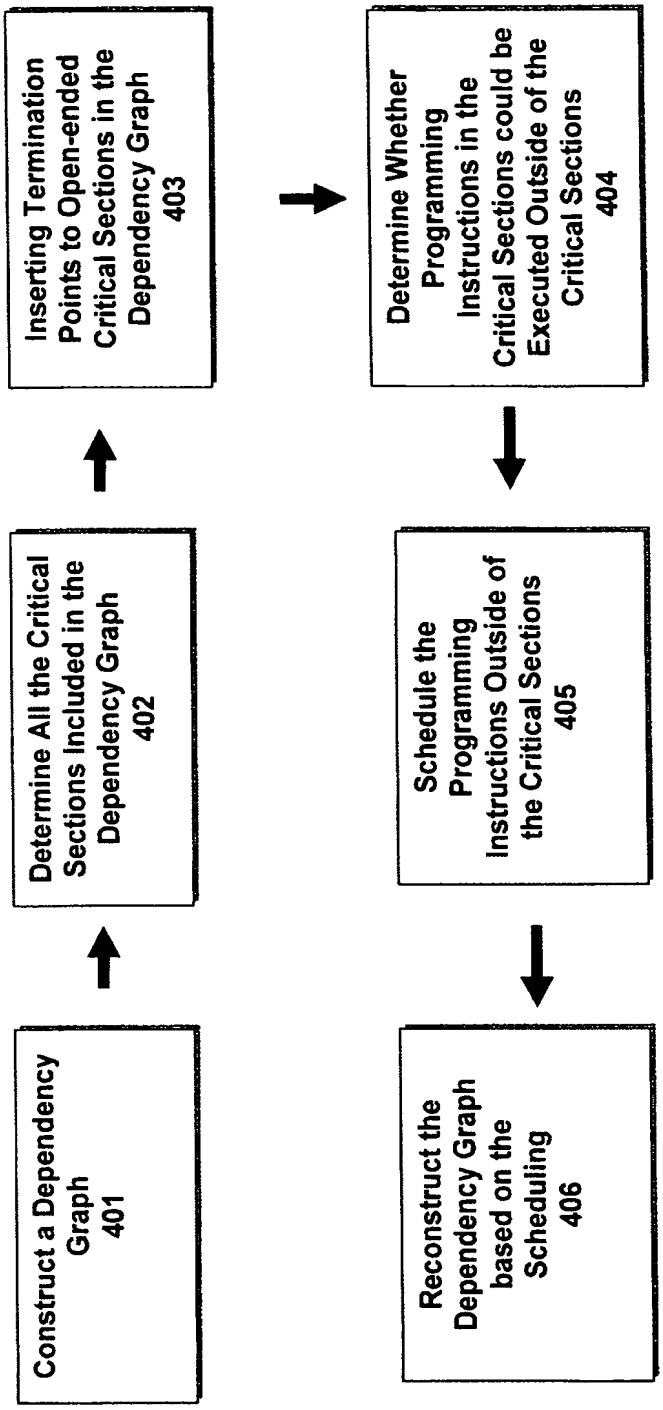

Construct a Dependency Graph 401
Determine All the Critical Sections Included in the Dependency Graph 402
Inserting Termination Points to Open-ended Critical Sections in the Dependency Graph 403
Determine Whether Programming Instructions in the Critical Sections could be Executed Outside of the Critical Sections 404
Schedule the Programming Instructions Outside of the Critical Sections 405
Reconstruct the Dependency Graph based on the Scheduling 406

Fig. 4

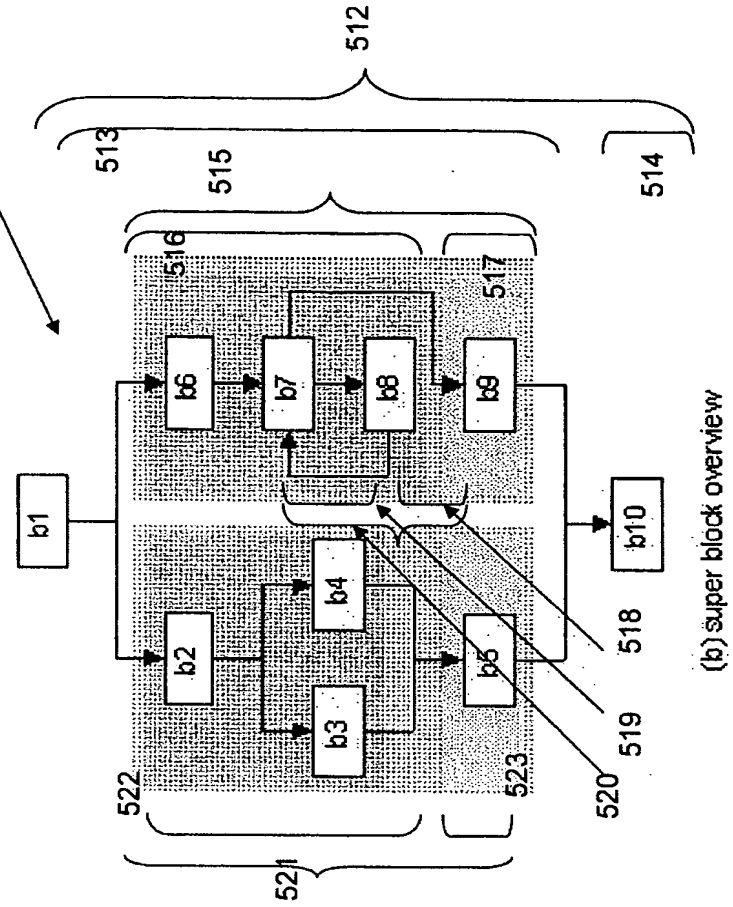

511
512
513
514
515
516
517
518
519
520
521
522
523
b1
b2
b3
b4
b5
b6
b7
b8
b9
b10
(b) super block overview

Fig. 5b

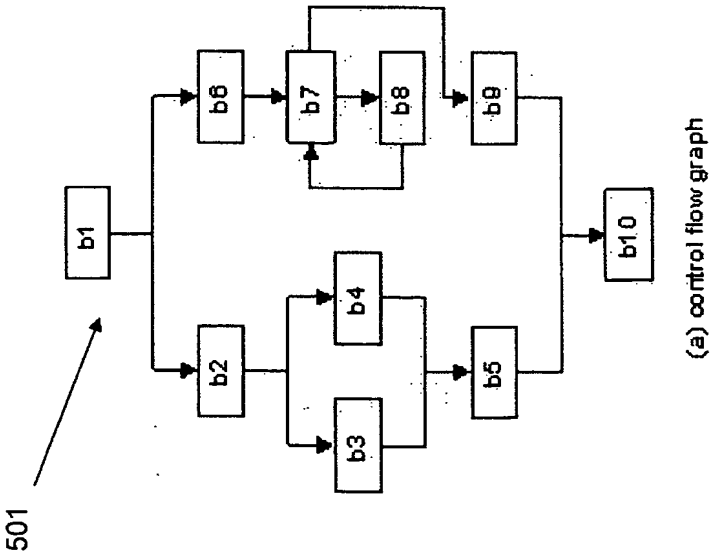
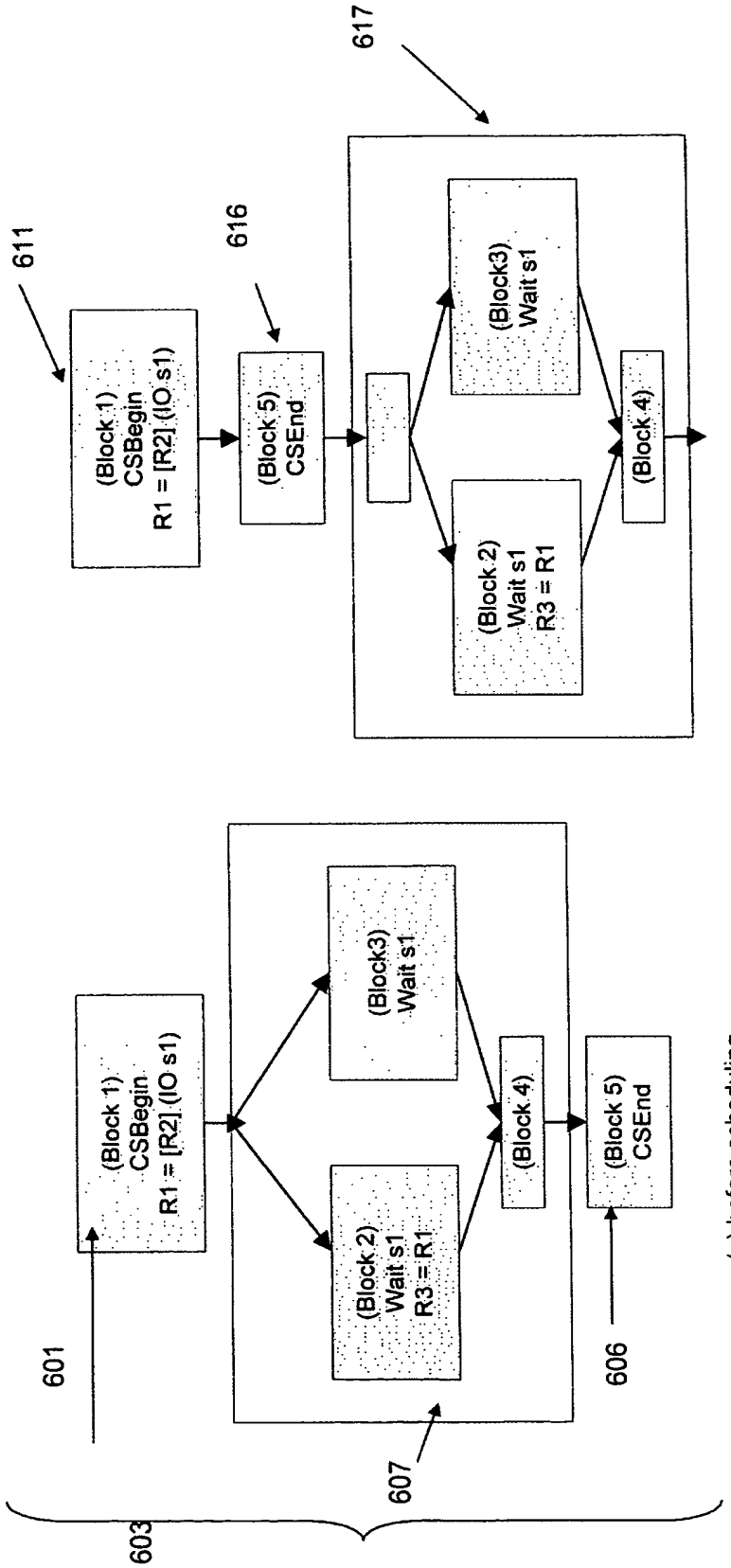

501
b1
b2
b3
b4
b5
b6
b7
b8
b9
b10
(a) control flow graph

Fig. 5a

Figs. 6a-b


603
601
(Block 1)
CSBegin
R1 = [R2] (IO s1)
(Block 2)
Wait s1
R3 = R1
(Block3)
Wait s1
(Block 4)
607
606
(Block 5)
CSEnd
(a) before scheduling
611
(Block 1)
CSBegin
R1 = [R2] (IO s1)
616
(Block 5)
CSEnd
617
(Block3)
Wait s1
(Block 2)
Wait s1
R3 = R1
(Block 4)
(b) after scheduling

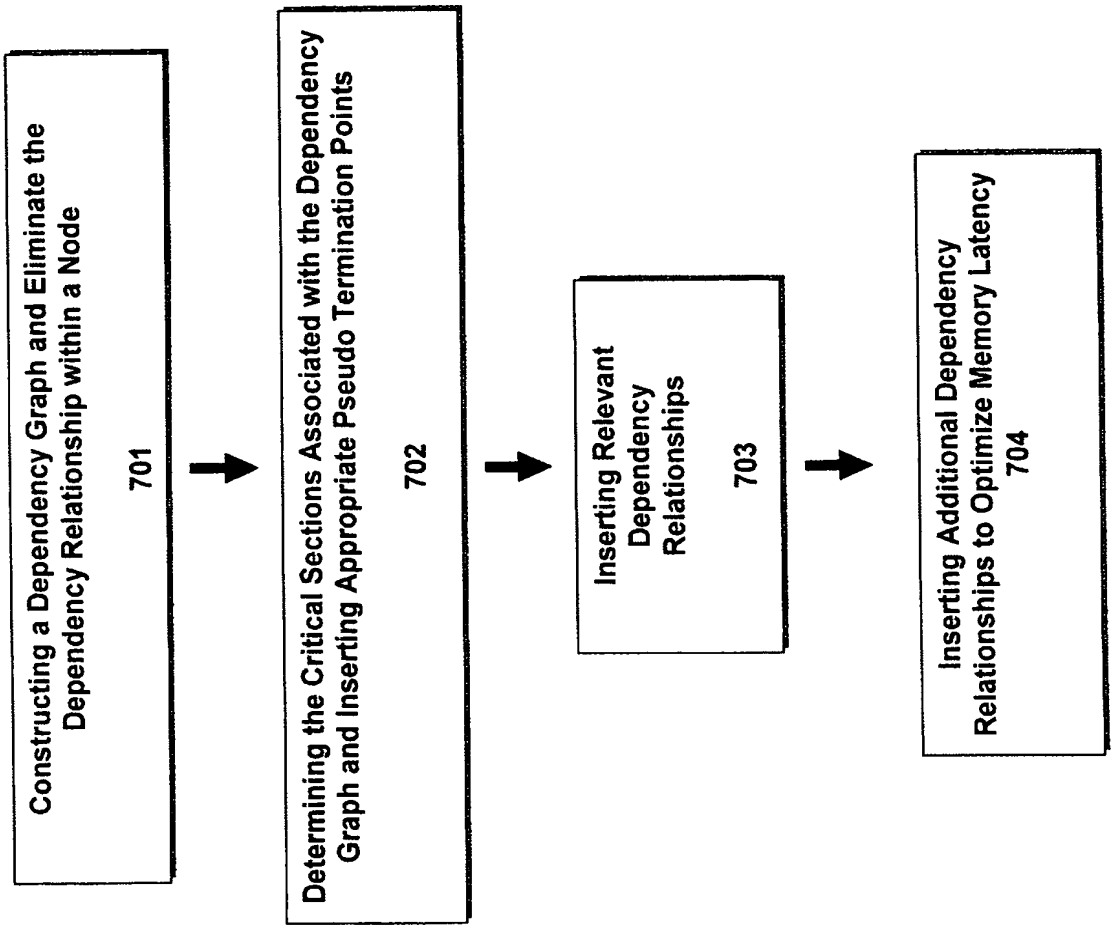
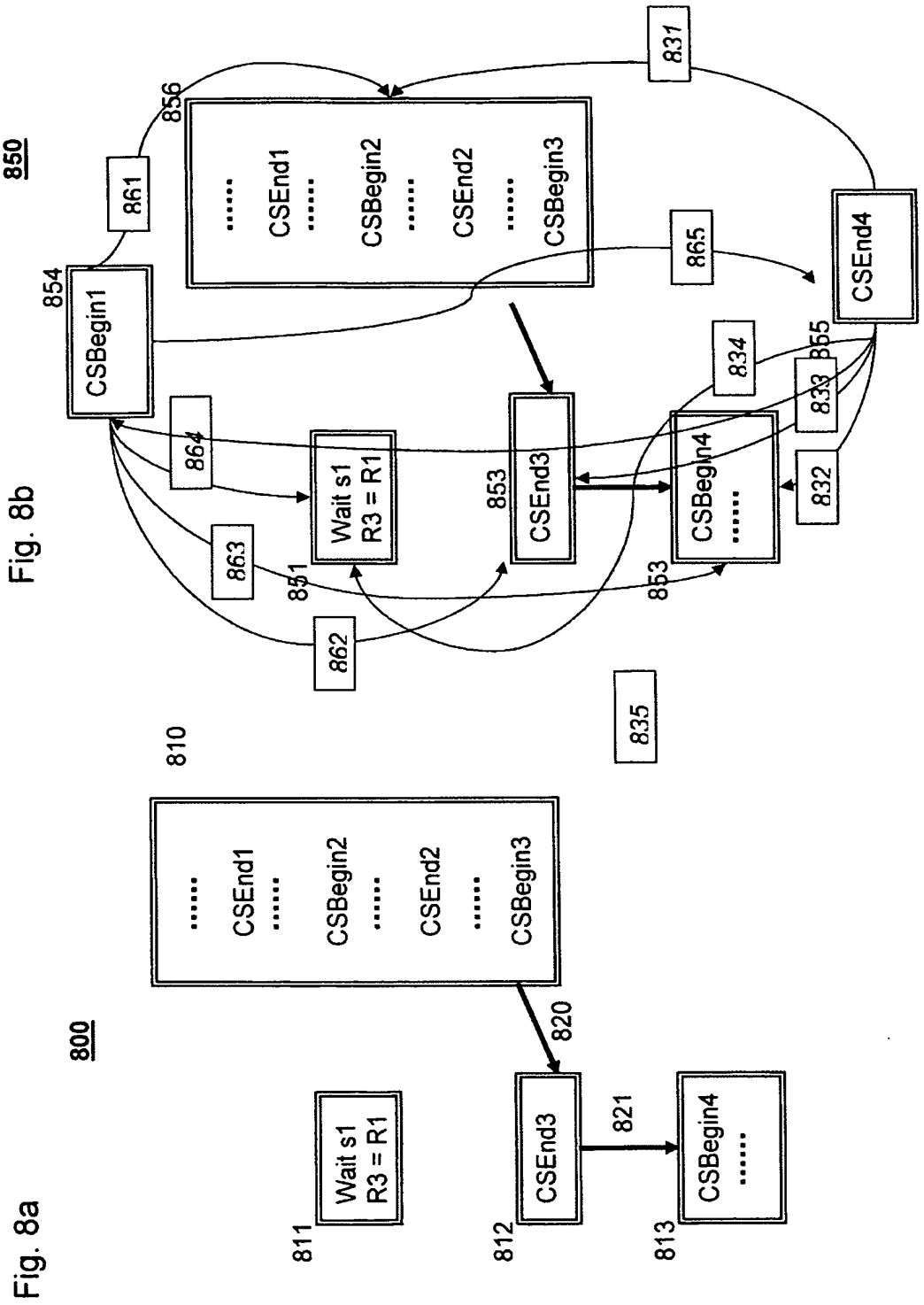

Constructing a Dependency Graph and Eliminate the Dependency Relationship within a Node
701
Determining the Critical Sections Associated with the Dependency Graph and Inserting Appropriate Pseudo Termination Points
702
Inserting Relevant Dependency Relationships
703
Inserting Additional Dependency Relationships to Optimize Memory Latency
704

Fig. 7


800
.....
CSEnd1
.....
CSBegin2
.....
CSEnd2
.....
CSBegin3
820
811
Wait s1
R3 = R1
812
CSEnd3
821
813
CSBegin4
.....
Fig. 8a
850
856
.....
CSEnd1
.....
CSBegin2
.....
CSEnd2
.....
CSBegin3
831
861
854
CSBegin1
865
CSEnd4
834
855
833
832
864
Wait s1
R3 = R1
853
CSEnd3
CSBegin4
.....
863
851
853
862
835
810
Fig. 8b

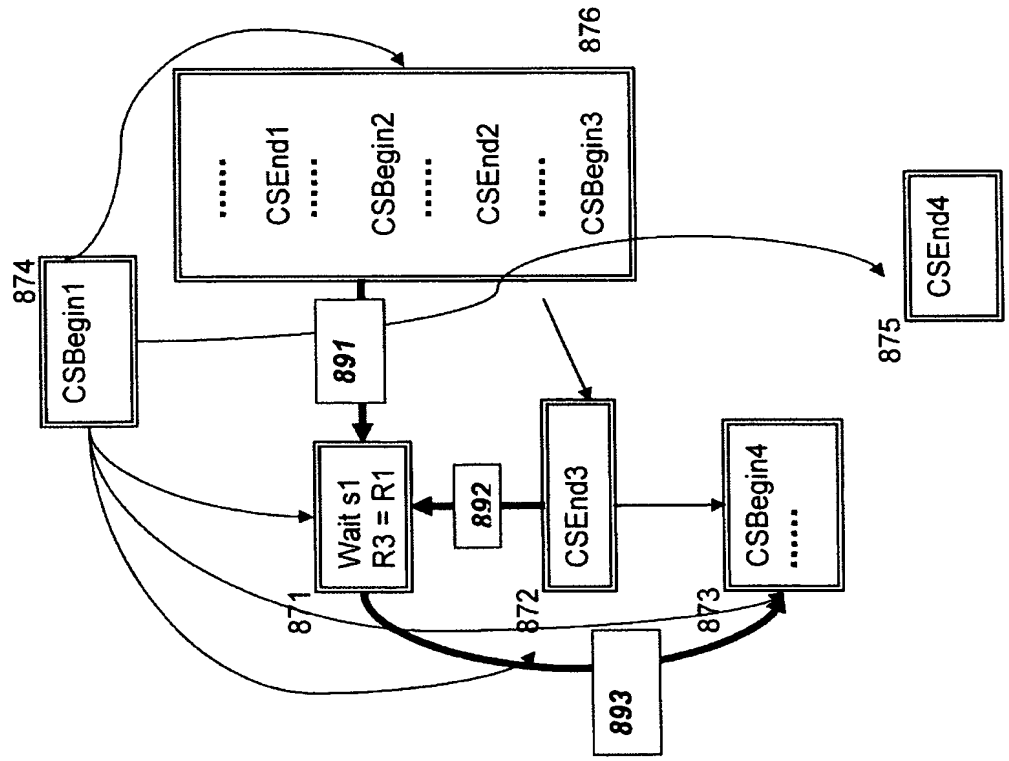

870
874
CSBegin1
876
.....
CSEnd1
.....
CSBegin2
.....
CSEnd2
.....
CSBegin3
891
875
CSEnd4
Wait s1
R3 = R1
892
CSEnd3
CSBegin4
.....
871
872
873
893

Fig. 8c